REMARKS

Appreciation is expressed for the courteous and helpful telephone interview granted by the Examiner on October 15, 1997 to Applicants' undersigned attorney and to Mr. George Shaw, a technical representative of Patriot Scientific Corporation, assignee of rights in the application. At the interview, the above changes to claim 71 in somewhat different language, the distinctions over the prior art and how those distinctions are defined by the claim language were discussed with the Examiner. It was agreed that these changes and discussion warranted further consideration of the application by the Examiner.

Claims 71-75 and 77-100 were rejected under 35 U.S.C. § 112, ¶ 2 as allegedly indefinite. Claims 71-75 and 77-100 were rejected under 35 U.S.C. § 103 as allegedly unpatentable over Boufarah in view of May, of record. These rejections are believed to be overcome by the above changes to the claims and the following remarks.

In the § 112 rejection, the language:

1. "instruction that accesses operands or instructions" in line 7 of claim 71,

2. "operands and instructions being located relative to said instruction groups" in line 8 of claim 71, and

3. "supplying means to supply to said central processing unit a remainder of said first of said instruction groups as said operand" in claim 77,

and similar language in other claims, is said to not be clear. As pointed out in the interview, in general, operands can be of many types, one aspect of the type determined by the addressing mode used to locate the operand. For instance, there can be register-addressed operands, direct-addressed memory operands, indirect addressed memory operands, or immediate operands. In the instant case, the only type of operands referred to are "immediate operands", that is, operands that are encoded as part of the instruction. On an 80x86 processor, for instance, the instruction MOV BX,#5 has one register operands and one immediate operand, the value 5 being encoded as part of the MOV instruction.

In the instant case, of concern is the manner in which an immediate operand or other instructions are located by an instruction when it is executed. The language "operands and instructions being located relative to said instruction groups" in line 8 of claim 71 and corresponding language in claims 91 and 97 has

been rewritten, both to make the language more definite and to define the invention better over the prior art, to specify more clearly that the operand and instructions are located relative to a boundary of the instruction groups. As pointed out in the interview, by establishing such a predefined location for the operand and instructions, processor construction is simplified and performance is enhanced because instructions and data are always optimally aligned. Operands are right justified within the instruction register so they are already properly aligned for use. Instructions are accessed by the location of the instruction group, thus conserving memory bandwidth by always obtaining a full instruction group of instructions to use, and allowing shorter branch instructions by reducing the number of address bits (to zero in the case of SKIPs and MLOOPs) required to address memory.

Thus from the above remarks the Examiner's concerns as stated: "It is not seen how instructions can be supplied as operands" and "...function of the instruction register and the instruction supplying means is not clear. Instruction register and instruction supplying means which are commonly for receiving and supplying instructions are recited for receiving and supplying operands" are overcome in that the operands of discussion are immediate operands and thus components of the instructions. There is no distinct standard term used in the industry to discuss the part of the instruction that is not the immediate operand, thus the term instruction is used to mean both the entire encoding including the immediate operand or a sub-part of the whole excluding the immediate operand.

As discussed in the interview, to enumerate the cases involved in "instruction that accesses operands or instructions" in line 7 of claim 71 (now reworded as "instruction that, when executed, causes an access to an operand or an instruction"), an instruction in the instruction group, when executed, can:

1. as exemplified by a SKIP instruction, cause access to another instruction or,

2. as exemplified by a LOAD-SHORT-LITERAL instruction, cause access to an operand, or

3. as exemplified by a BRANCH instruction cause access to both by accessing an operand for the destination addressing information and another instruction group as the destination.

The above changes to this language make it clear that execution of the instruction results in these accesses.

As explained in the interview, claim 72 in fact specifies the manner in which a SKIP instruction operates in this invention to always immediately go to the first instruction of the next instruction group, thus not executing, or "skipping", the remainder of the current instruction group. Additionally, this language is further clarified by the amendment to claim 71 specifying instructions being accessed are located relative to a boundary of the instruction groups. Again, this manner of operation for a SKIP instruction is not conventional since a varying number of instructions might be skipped depending upon where the programmer had placed the SKIP within the current instruction group, and no addressing bits are used in the SKIP to vary the number of instructions.

Similarly, claim 74 specifies that a MICROLOOP instruction always goes back to the first instruction in the instruction register (that is, the first instruction of the current group). Similar to SKIP, this operation eliminates the need to specify a destination address for a MICROLOOP and the number of instructions looped depends on where the programmer had placed the MICROLOOP within the current instruction group. In accordance with parent claim 71 as amended, this first instruction is at the boundary of the instruction group.

Claims 73 and 75 have been rewritten to specify that the SKIP or MICROLOOP instructions are executed or not executed based on existence of the predefined condition.

Comparable changes have been made in claims 91-100 as appropriate. The rejection under 35 U.S.C. § 112 is believed to be overcome.

In addition to overcoming the rejection under § 112, the above changes to the claims also differentiate them more clearly over the Boufarah et al. and May references of record, whether considered separately or in combination. Neither Boufarah et al. nor may teach or suggest "operand or instruction being located relative to a boundary of said instruction groups", as now recited in claims 71, 91 and 97, nor the resulting simplification of the instruction set implementation. Neither Boufarah et al. nor May teach or suggest the operation of a SKIP instruction to supply the first instruction from the next instruction group, as recited in claims 72 and 92. Neither Boufarah et al. nor May teach or suggest the operation of a

MICROLOOP instruction to supply the first instruction in the instruction register (the first instruction of the current group), as recited in claims 74 and 94. The operation of the SKIP and MICROLOOP instructions in this manner means that no destination addresses for those operations need to be included in the instruction even though the number of instructions skipped or looped can vary (by instruction placement). At least for that reason, the claimed subject matter is far more than a matter of design choice. For these reasons, the rejection under § 103 is believed to be overcome.

Based on the above changes to the claims and remarks, all of the claims in the application are believed to be patentable over the prior art. This application is believed to be in condition for allowance, and allowance is solicited.

Respectfully submitted,

COOLEY GODWARD LLP

Willis E. Higgins
Reg. No. 23,025

Five Palo Alto Square
Palo Alto, CA 94306-2155
Telephone: (650) 843-5145